

## Re: Confusion about splitting classes to allow sharing of resources

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.object/2005-04/msg00438.html>

---

- *From:* "m.a.stijnman@xxxxxxxxxxxxxxxx" <m.a.stijnman@xxxxxxxxxxxxxxxx>
  - *Date:* 20 Apr 2005 15:14:08 -0700
- 

H. S. Lahman wrote:

> Responding to M.a.stijnman...

>

>> I've been confusing myself with this problem for a bit, and I'm

>> wondering if I'm going about this the right way. Let me describe my

>> problem first:

>>

>> I'm using cubic splines to model two-dimensional parametric curves

>>  $(x(t), y(t))$  to interpolate between nodes. I have a working Spline

class

>> that does cubic spline interpolation of function values  $f[i]$  that

are

>> defined at knot positions  $t[i]$ . I use this in my current Curve

class.

>> Along with the splines for  $x$  and  $y$ , the Curve class also has one

for

>> any other values (say  $p$  and  $q$ ) that need to be interpolated along

the

>> curve. I expect subclasses of Curve to add new interpolated

parameters

>> along the curve. All Splines  $x$ ,  $y$ ,  $p$  and  $q$  currently have their own

>> knot vector  $t[i]$  and the Curve class provides functions that ensure

a

>> certain  $t[i]$  of all splines only get changed all at once (in my app

the

>>  $t[i]$  sometimes need to change to enhance the parameter mapping of

the

>> curve). This is done by providing a virtual SetKnot function that a

>> Curve implementation should override. To me, the logical next step

>> would be, to change the spline class so it supports sharing a knot

>> vector between splines – it will save storage and will guarantee

all

>> Splines stay in sync. For this, I would add a class KnotVector and

give

>> the Spline class a reference to a (possibly) external knot vector.

>> Adding new interpolation variables in subclasses of Curve will then

>> just be a matter to linking that spline to the same knot vector as

>> well.

Re: Confusion about splitting classes to allow sharing of resources

>  
> I don't know much about spline interpolation algorithms, so I am  
> guessing a bit here. I assume there is a different interpolation  
> algorithm for p and q than for x and y. I assume that 'node' is the  
> curve viewpoint and a node is always a position (x, y) but may have  
> other arbitrary parameters (p, q, ...) and that 'knot' is the  
> interpolation algorithm viewpoint.  
>  
> If so, what you have described seems to be:  
>  
> 1 interpolates for \*  
> [2DCurve] ----- [Spline]  
> | 1 R1 A  
> R2 || R3  
> | composed of +-----+-----+  
> | \* ||  
> [Node] -----+ [XandY] [PandQ]  
> + x 1 corresponds to || 1 | 1  
> + y | R4 || R6  
> | 1 | R5 | interpolates | interpolates  
> | associated with || between | between  
> R7 || 1 | \* | \*  
> | +----- [XYKnot] [PQKnot]  
> || 1  
> | \* R8 |  
> [Parameter] -----+  
> + p 1 corresponds to  
> + q  
>  
> The Knot vectors are represented by the R5 and R6 relationships and  
> the  
> curve node vector is represented by the R2 relationship. By  
> separating  
> the Parameter values, one can add parameters by simply dynamically  
> instantiating the R7 relationship (assuming an appropriate algorithm  
> exists in a [Spline] subclass). [Note that [Spline] here is  
> essentially  
> the GoF Strategy pattern.]  
>  
> Synchronization is maintained over the R4 and R8 relationships  
> whenever  
> there is a change.  
>  
>>  
>> Now for my dilemma: my current spline class supports a public  
> member  
>  
> I think this is the problem. Managing the relationships is really an  
  
> instantiation problem for objects and relationships. Once that is  
> done,

Re: Confusion about splitting classes to allow sharing of resources

- > the navigation will always "walk" the right vectors during the
- > computations. IOW, I think the "add" capability is a collection
- class
- > responsibility that should be managed by a factory rather than the
- curve
- > or interpolation classes. That is, it is a structural definition
- that
- > is orthogonal to the computational problem.
- >
- > When a Curve is initially instantiated one needs to (a) instantiate a
  
- > Curve, (b) instantiate the relevant Nodes and add them to the R5
- > collection class, (c) instantiate corresponding XYKnot objects and
- the
- > R4/R5 relationships, (d) instantiate any relevant Parameter objects
- and
- > add them to the R7 collection, and (e) instantiate the corresponding
- > xxKnot objects and the relevant relationships (e.g., R6/R8).
- >
- > Note that (d) and (e) can be done dynamically at run time; the
- > processing is exactly the same as it they were "hard-wired" at
- startup.
- > Also note that all this instantiation can be defined in external
- > configuration data. So one could add or remove nodes and parameters
- to
- > be interpolated or change algorithm assignments without touching the
- > code. However, that assumes that an appropriate subclass of [Spline]
  
- > exists that contains the right algorithm. To add a new algorithm one
  
- > would still have to muck with the code.
- >
- >
- > \*\*\*\*\*
- > There is nothing wrong with me that could
- > not be cured by a capful of Drano.
- >
- > H. S. Lahman
- > hsl@xxxxxxxxxxxxxxxxxxxx
- > Pathfinder Solutions -- Put MDA to Work
- > <http://www.pathfindermda.com>
- > blog: <http://pathfinderpeople.blogs.com/hslahman>
- > (888)OOA-PATH

First of all, thanks for the elaborate answer, you gave me a few things to think about. I think I also see one of the flaws in my thinking here. Let me explain a little on how I have a spline implemented now (without going into the mathematics). Basically, it holds a vector of strictly increasing knots  $t[i]$ , and a vector of corresponding function values  $f[i]$ . It can also act as a function  $f(t)$ , which returns the interpolated values at a certain  $t$ . Also, it can calculate an

Re: Confusion about splitting classes to allow sharing of resources

approximate derivative  $f'(t)$ . Indeed one can imagine different interpolation schemes, like linear and cubic interpolation, using the same interface. I think where my thinking got flawed, is that I use spline objects to actually \*store\* the data that needs interpolation. Whenever some of the data is changed, the spline object sets a flag so that when next time interpolation is performed, the 'behind-the-scenes' interpolation data gets regenerated. What you are telling me here, if I understand correctly, is I should think of spline as an algorithm, rather than as an object. Probably my data to be interpolated should then also be stored outside of the spline instance. Is that right?

I think the schema can be a little simplified in my case, since both the position (x,y) as the parameters p and q are modelled as a parametric function of a single parameter t. The exact shape of the function is determined by the values of x, y, p and q at the curve nodes, as well as the value of t defined at the curve nodes. This vector  $t[i]$  is what I refer to as the knot vector and is what is used to build the splines. So there is only one knot vector, even though each variable can have it's own type of interpolation.

As for the adding of nodes and knots, you're saying that should be handled by whatever class wants to instantiate a Curve or a Spline, right? Especially for the splines that makes much more sense if I see the spline class as an algorithm on outside data, not as an object holding data. So I should write a SplineFactory class, and a CurveFactory class, to provide that sort of facilities, correct? I think I'll read up on factory classes a bit then, figure out how best to implement them.

best regards Mark

---

• ***Follow-Ups:***

- ◆ ***Re: Confusion about splitting classes to allow sharing of resources***

◇ *From:* H. S. Lahman

• ***References:***

- ◆ ***Confusion about splitting classes to allow sharing of resources***

◇ *From:* m.a.stijnman@xxxxxxxxxxxxxxxx

- ◆ ***Re: Confusion about splitting classes to allow sharing of resources***

◇ *From:* H. S. Lahman

- Prev by Date: ***Re: Looking for UML Diagrams exercises + solutions***
- Next by Date: ***Re: Is there a BEA Tuxedo equivalence in Java?***
- Previous by thread: ***Re: Confusion about splitting classes to allow sharing of resources***
- Next by thread: ***Re: Confusion about splitting classes to allow sharing of resources***
- Index(es):
  - ◆ ***Date***

Re: Confusion about splitting classes to allow sharing of resources

◆ *Thread*